

# Location Estimation Using Crowdsourced Geospatial Narratives

Georgios Skoumas

School of Electrical and Computer Engineering, NTUA  
gskoumas@dblab.ece.ntua.gr

Dieter Pfoser

Dept. of Geography and Geoinformation Science, GMU  
dpfoser@gmu.edu

Anastasios Kyrillidis

School of Computer and Communication Sciences, EPFL  
anastasios.kyrillidis@epfl.ch

August 27, 2014

## Abstract

The “crowd” has become a very important geospatial data provider. Subsumed under the term Volunteered Geographic Information (VGI), non-expert users have been providing a wealth of quantitative geospatial data online. With spatial reasoning being a basic form of human cognition, narratives expressing geospatial experiences, e.g., travel blogs, would provide an even bigger source of geospatial data. Textual narratives typically contain qualitative data in the form of objects and spatial relationships. The scope of this work is (i) to extract these relationships from user-generated texts, (ii) to quantify them and (iii) to reason about object locations based only on this qualitative data. We use information extraction methods to identify toponyms and spatial relationships and to formulate a quantitative approach based on distance and orientation features to represent the latter. Positional probability distributions for spatial relationships are determined by means of a greedy Expectation Maximization-based (EM) algorithm. These estimates are then used to “triangulate” the positions of unknown object locations. Experiments using a text corpus harvested from travel blog sites establish the considerable location estimation accuracy of the proposed approach.

## 1 Introduction

User-contributed content has benefited many scientific disciplines by providing a wealth of new data sources. In the geospatial domain, authoring content typically involves quantitative, coordinate-based data. While technology has helped a lot to facilitate geospatial data collection, e.g., all smart phones are equipped with GPS positioning sensors, yet authoring quantitative data requires specialized applications (often part of social media platforms) and/or specialized knowledge, e.g., Openstreetmap<sup>1</sup>. This fact hinders the widespread adoption of VGI as an even bigger, large-scale geospatial data source.

The broad mass of users contributing content on the Internet are much more comfortable using *qualitative information*. People typically do not use coordinates to describe their spatial experiences (trips, etc.), but rely on qualitative concepts in the form of toponyms (landmarks) and spatial relationships (near, next, etc.). Crowdsourcing qualitative geospatial data is thus more challenging since the mental model of space is actually very different from the representation that is used to record datasets.

Utilizing qualitative geospatial data is typically based on trying to quantify it. One of the challenges here is the uncertainty associated with the data. The same concept (near) might be interpreted differently by the various users. As an example, consider the following narrative. “*The best pita place in Greece is **next to** the Monastiraki Metro Station in Athens.*” In this case, we want to quantify what people imply when they say “*next to*”. Being able to do so, might allow us to actually discover the “best pita place in Greece”. Eventually, by collecting more observations that mention the “best pita place in Greece” using qualitative spatial information, i.e., spatial relationships tying the place to known locations, we will be able to refine the unknown location and, thus, locate places that otherwise could not be geocoded.

To this end, we consider the following problem:

**PROBLEM:** *Given a set of objects  $P_K$  with a-priori known coordinates in space, a set of objects  $P_U$  whose exact positions are unknown, and a set of observed spatial relationships  $R$  between objects of set  $P_U$  and objects of set  $P_K$ , find probabilistic estimates of the positions of objects in the set  $P_U$ , based on their observed spatial relationships  $R$  with objects in the set  $P_K$ .*

Even though the above formalization is very comprehensive, the problem contains high uncertainty, especially when the source of spatial information is user-contributed. To achieve high accuracy location estimates, our approach follows a probabilistic path, where we quantify qualitative relations as probability measures. Essentially, using textual narratives, we observe

<sup>1</sup><https://www.openstreetmap.org/>

point-of-interest (POI) pairs that are linked together by a spatial relationship. Assuming that both locations are known, each observation is roughly quantified using a spatial feature vector comprising distance and orientation. We use information extraction methods to identify those toponyms and spatial relationships in texts. A greedy Expectation Maximization-based (EM) method is used to train a probability distribution, which represents the quantified spatial relationships under a probabilistic framework. Given a specific spatial relation, it provides a set of random variables (spatial feature vector) that have certain probability density functions (PDFs) associated with them, for a specific spatial relation. These positional PDFs are then used to “triangulate” the positions of unknown POI locations. The more observations we have with respect to an unknown location, the preciser we will be able to reason about the POI’s unknown location. Actual location estimation experiments using textual narratives from travel blogs establish the validity and quality of the proposed approach.

The outline of the remainder of this work is as follows. Section 2 discusses related work. Section 3 discusses the specific qualitative data involved and introduces the spatial feature vectors used for quantification, while Section 4 introduces the tools necessary to derive quantification in the form of PDFs for the spatial relationships. Section 5 validates the proposed approach using synthetic and real world location estimation scenarios. Finally, Section 6 presents conclusions and directions for future work.

## 2 Related Work

Work relevant to this research includes: (i) extraction of semantic or especially spatial relations from natural language expressions, (ii) qualitative modeling of spatial relations and its application to spatial databases, and, (iii) quantitative modeling of spatial knowledge.

The extraction of qualitative spatial data from texts requires the utilization of efficient natural language processing (NLP) tools to automatically extract and map phrases to spatial relations. In the past, extraction of *semantic* relations between entities in texts is developed in [3, 9, 25, 19, 29], while extraction of *spatial* relations between entities in texts is analyzed in [15, 28, 30]. While the above works constitute a good match in our developments for spatial relationship extraction from texts, we intentionally designed our specialized qualitative spatial data mechanism that better fits into the particularity, e.g. noisy crowdsourced data, of the relation extraction part of our problem.

Formal methods for qualitative representation of spatial relationships based on mathematical theories of order are presented in [6, 7, 8, 13]. Their applicability on spatial database systems and some key-role technical concepts are coherently discussed in [10, 21]. Qualitative representation of spatial knowledge is discussed in [16, 20] where the authors identify the common concepts of the qualitative representation and processing of spatial knowledge. In this work, we bridge the gap between qualitative and quantitative data by quantifying qualitative spatial relationships extracted from user generated texts.

Recent research on *quantitative representation of spatial knowledge* has been conducted in relation to situational awareness systems, robotics, and image processing. Modeling uncertain spatial information for situational awareness systems is discussed in [14, 18]. The authors propose a Bayesian probabilistic approach to model and represent uncertain event locations described by human reporters in the form of free text. Estimation of uncertain spatial relationships in robotics is addressed in [23]. A probabilistic algorithm for the estimation of distributions over geographic locations is proposed in [11] where a data-driven scene matching approach is used in order to estimate geographic information based on images. Image similarity based on quantitative spatial relationship modeling is addressed in [26] while, in [27], a fuzzy decision tree algorithm is proposed to formalize spatial relations between linear objects. Finally, in [22], the authors introduce a basic Expectation Maximization/-Gaussian Mixture Model approach to quantify qualitative spatial data manually extracted from texts. In this work, we provide a framework for the automatic extraction of qualitative spatial data from texts and we introduce a location estimation method based on spatial relation fusion. This bridges the gap between qualitative and quantitative representation of spatial relations using efficient machine learning techniques and arrives at an actual text-to-map application: starting from extraction, moving to modeling and finally to location estimation.

## 3 Qualitative Spatial Data

This section highlights our approach on qualitative data extraction from texts and presents a model for representing spatial relationships based on distance and orientation measures.

### 3.1 Textual Narratives as Data Source

Crowdsourced narratives are likely to contain spatial information, if we focus on text that is related to spatial experiences. In this work, we choose travel blogs as a rich potential geospatial data source. This selection is based on the fact that people tend to describe their experiences in relation to their trips and places they have visited, which results in “spatial” narratives. To gather such data, we use classical Web crawling techniques [5] and compile a database<sup>2</sup> consisting of 120,000 texts, obtained from travel blogs<sup>3</sup>.

---

<sup>2</sup>Available upon request

<sup>3</sup>TravelBlog, TravelJournal, TravelPod

### 3.2 Spatial Relations

Obtaining qualitative spatial relations from text involves the detection of (i) spatial objects, i.e., Points-of-Interest (POIs) or toponyms and (ii) spatial relationships linking the POIs. The employed approach involves geoparsing, i.e., the detection of candidate phrases, and geocoding, i.e., linking the phrase/toponym to actual coordinate information.

Using the Natural Language Processing Toolkit (NLTK) (cf. [1]), which is a leading platform to analyze raw natural language data, we managed to extract 500,000 POIs from the text corpus. For the geocoding of the POIs, we rely on the GeoNames<sup>4</sup> geographical gazetteer data, which covers all countries and contains over ten million place names and their coordinates. This procedure associates (whenever possible) geographic coordinates with POIs found in the travel blogs, using string matching based on the Levenshtein string distance metric (cf. [12]). Using the GeoNames gazetteer we managed to geocode about 480,000 out of the 500,000 extracted POIs.

Having identified and geocoded the spatial objects, the next step is the extraction of qualitative spatial relationships. As mentioned in Section 2, the extraction of spatial relations between entities in text is a hard NLP problem. REVERB (i.e. [9]) and EXEMPLAR (i.e. [19]) are the state of the art available software tools for the extraction of semantic relations between identified entities in texts. The main drawback of these approaches is that they are not trained for spatial relations specifically. Moreover, we observe that they might perform poorly when applied with a noisy crowdsourced dataset.

We address this NLP challenge by implementing a spatial relation extraction algorithm based on NLTK [1] components in combination with predefined strings and syntactical patterns. More specifically, we define a set of language expressions that are typically used to express a spatial relation in combination with a set of syntactical rules. The use of both syntactical and string matching reduces the number of false positives considerably. As an example, consider the following phrase. “*Deutsche Bank invested 10 million dollars in Brazil.*”. Here, a simple string matching solution would extract a triplet of the form (Deutsche Bank, in, Brazil), which is a false positive. In our approach, the use of predefined syntactical patterns avoids this kind of mistakes. On the other hand, for the phrase “*Deutsche Bank invested 10 million dollars in Rio de Janeiro, which is in Brazil.*” our algorithm would extract a triplet of the form (Rio de Janeiro, in, Brazil) which is a true positive.

In Table 1, we show a spatial relation extraction task on a small dataset of 300 annotated crowdsourced spatial relations. Both our method and EXEMPLAR perform better than REVERB in terms of precision and recall. While our NLTK approach seems to have a slightly lower precision than EXEMPLAR [19], it achieves a really higher recall, where the fraction of extracted relations relevant to the query is higher. This is the reason we preferred to use our NLTK-based spatial relation method instead of EXEMPLAR.

Table 1: Precision and recall for three different spatial relation extraction approaches.

Method	Precision	Recall
EXEMPLAR [19]	<b>0.7</b>	0.4
REVERB [9]	0.1	0.4
NLTK	0.6	<b>0.82</b>

Algorithm 1 describes the architecture of the proposed information extraction system. Initially, the raw text document is segmented into sentences (Step 3). Each sentence is further subdivided (tokenized) into words and tagged as part-of-speech (Steps 5-6). Continuing name entities (POIs) are identified (Step 7). We will typically be looking for relations between specified types of named entities, which in NLTK are *Organizations*, *Locations*, *Facilities* and *Geo-Political Entities* (GPEs). In sequence, in case there are two or more name entities in the sentence, we check if any of the predefined syntactical patterns applies between the recognized name entity pairs (Step 12). If it applies, we then use regular expressions to determine the specific instance of the observed spatial relation from our predefined spatial relation pattern list for this case (Step 14). If there is a string pattern match we record the extracted triplet (Steps 15-18). Thus, the search for spatial relations in texts results into a set of triplets  $O$  of the form  $(P_u, R_o, P_v)$ , where  $P_u$  and  $P_v$  are named entities of the required types and  $R_o$  is the observed spatial relation that intervenes between  $P_u$  and  $P_v$ .

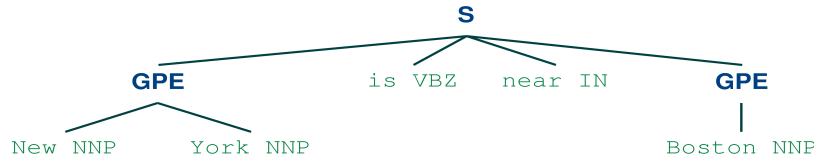


Figure 1: Example of a parsed sentence syntactic tree.

A relation extraction example is shown in Figure 1, where the sentence is analyzed as explained and two named entities are

<sup>4</sup><http://www.geonames.org/>

---

**Algorithm 1:** Spatial Relation Extraction

---

**Input:** A database of texts  $T$ , a set of syntactical patterns  $A$ , a set of spatial relation string patterns  $R$

**Output:** A set of triplets  $O$  of the form  $(P_u, R_o, P_v)$  where  $P_u \neq P_v$  and  $R_o \in R$

```
1 begin
2   foreach text  $t \in T$  do
3     Extract sentences from  $t$  into set  $S$ 
4     foreach sentence  $s \in S$  do
5       Token  $s$  using NLTK
6       PosTag  $s$  using NLTK
7       Identify name entities using NLTK
8       if two or more name entities in  $s$  then
9         Extract POI pairs in  $P$ 
10        foreach  $p \in P$  do
11           $p_A \leftarrow$  Extract syntactical pattern of  $p$ 
12          if  $p_A \in A$  then
13             $p_R \leftarrow$  Extract string pattern of  $p$ 
14            if  $p_R \in R$  then
15               $P_u \leftarrow p(1)$ 
16               $P_v \leftarrow p(2)$ 
17               $R_o \leftarrow p_R$ 
18               $O.PUSHTRIPLET(P_u, R_o, P_v)$ 
19            end
20          end
21        end
22      end
23    end
24  end
25  return  $O$ 
26 end
```

---

identified as GPEs. We first check the syntax and make sure that the pattern “GPE - 3rd person verbal phrase (VBZ) - preposition/-subordinating conjunction (IN) - GPE” exists in our set of predefined spatial relation patterns. Performing string matching on the intermediate chunks (“near”) results in the triplet (*New York, Near, Boston*).

Applying Algorithm 1, we extracted 440,000 triplets from our 120,000 travel blog text corpus. Figure 2 shows a small sample of a *Spatial Relationship Graph*, i.e., a spatial graph in which nodes represent POIs and edges label spatial relationships existing between them. The graph visualizes a sample of the spatial relationship data collected for New York city. In this work we extracted spatial relation data for four different cities, i.e. London, New York, Paris and Beijing. These four cases, going gradually from sparse to very dense spatial relation data, will be our main datasets during the experimental evaluation of the proposed approach.

### 3.3 Spatial Features

Statistical models are often used to represent observations in terms of random variables. These models can then be used for estimation, description, and prediction based on basic probability theory. In our approach, we model a spatial relation between two POIs  $P_u, P_v$  in terms of *distance* and *orientation*. We consider a labeled *Spatial Feature Vector* as two random variables that model spatial relations in a probabilistic way.

Assuming a projected (Cartesian) coordinate system, the distance is computed as the Euclidean metric between the two respective coordinates. The orientation is established as the counterclockwise rotation of the x-axis, centered at  $P_v$ , to point  $P_u$ .

Several instances of a spatial relation are used to create a dataset, which will be used to train one probabilistic model for each spatial relation. For a concise and consistent mathematical formalization, let us consider that for each instance of each relation, we create a two-dimensional spatial feature vector  $X = (X_d, X_o)^T$  where  $X_d$  denotes the distance and  $X_o$  denotes the orientation between  $P_u$  and  $P_v$ . This way, we end up with a set of two-dimensional feature vectors  $\mathcal{X} = \{X_1, X_2, \dots, X_n\}$  for each spatial relation.

## 4 Spatial Relationship Modeling

In this section we describe the probabilistic modeling we follow in order to quantify qualitative spatial data. Key ingredients of our system are methods that train probabilistic models. Our analysis below includes (i) a short description of the probabilistic mixture models we employ for the quantitative representation of spatial relations (Section 4.1) and, (ii) a greedy learning

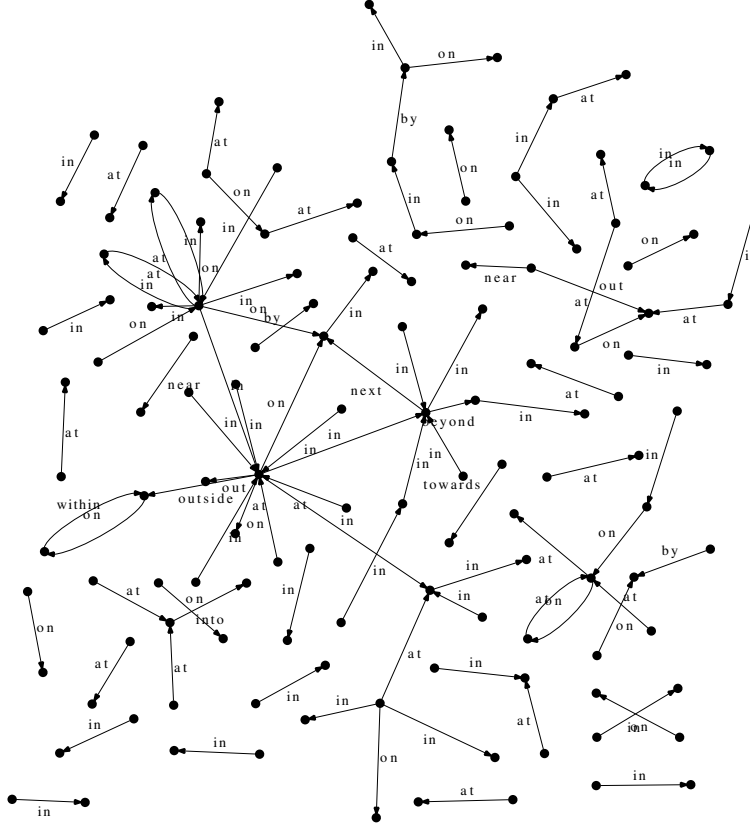


Figure 2: Small sample of a spatial relation graph for New York city.

algorithm for model parameter estimation (Section 4.2). Overall, this section describes a method that trains probabilistic models that quantify crowdsourced spatial relationships. These estimates can then be used to reason about unknown POI locations in textual narratives (see Section 5).

#### 4.1 Quantifying Qualitative Relations

An essential step in quantifying qualitative data is the mapping of the generated data to pre-selected probability density functions (PDFs). In [17], it is shown that for any heterogeneous multi-dimensional data that originates from an *arbitrary* PDF  $p$ , there exists a sequence of finite mixtures  $p_k(x) = \sum_{i=1}^k w_i g(x; \theta_i)$  that achieves Kullback-Leibler (KL) divergence

$$D(p||p_k) - D(p||g_p) \leq \mathcal{O}(1/k)$$

for any  $g_p = \int g(x; \theta) P(d\theta)$ . i.e., one can achieve a good approximation of  $p$  with rate  $\mathcal{O}(1/k)$  by using a  $k$ -component mixture of  $g(x; \cdot)$ . Furthermore, this bound is achievable by employing a greedy training scheme [17], i.e., we can approximate any density  $p$  by a greedy training procedure.

In this work we employ Gaussian Mixture Models (GMMs) which have been extensively used in many classification and general machine learning problems (cf. [2]). They are very well known for (i) their formality, as they build on the formal probability theory, (ii) their practicality, as they have been implemented several times in practice, (iii) their generality, as they are capable of handling many different types of uncertainty, and (iv) their effectiveness.

In general, a GMM is a weighted sum of  $M$  component Gaussian densities as  $p(x|\lambda) = \sum_{i=1}^M w_i g(x; \mu_i, \Sigma_i)$  where  $x$  is a  $d$ -dimensional data vector (in our case  $d = 2$ ),  $w_i$  are the mixture weights, and  $g(x; \mu_i, \Sigma_i)$  is a Gaussian density function with mean vector  $\mu_i \in \mathbb{R}^d$  and covariance matrix  $\Sigma_i \in \mathbb{R}^{d \times d}$ . To fully characterize  $f$ , one requires the mean vectors, the covariance matrices and the mixture weights. These parameters are collectively represented in  $\lambda = \{w_i, \mu_i, \Sigma_i\}$  for  $i = 1, \dots, M$ .

In our setting, each spatial relation is modeled by a 2-dimensional GMM, trained on each relation's spatial feature vectors, as detailed in Section 3.3. For the parameter estimation of each Gaussian component of each GMM, we use Expectation Maximization (EM) (cf. [4]). EM enables us to update the parameters of a given  $M$ -component mixture with respect to a feature vector set  $\mathcal{X} = \{X_1, \dots, X_n\}$  with  $1 \leq j \leq n$  and all  $X_j \in \mathbb{R}^d$ , such that the log-likelihood  $\mathcal{L} = \sum_{j=1}^n \log(p(X_j|\lambda))$  increases with each re-estimation step, i.e., EM re-estimates model parameters  $\lambda$  until  $\mathcal{L}$  convergence.

## 4.2 Model Optimization

A main issue in probabilistic modeling with probability mixtures is that a predefined number of components is neither a dynamic nor an efficient and robust approach. The optimal number of components should be decided based on each dataset.

We employ a greedy learning approach to dynamically estimate the number of components in a GMM. (cf. [24]). Our approach builds the mixture component in an efficient way by starting from an one-component GMM—whose parameters are trivially computed by using EM—and then employing the following two basic steps until a stopping criterion is met:

1. Insert a new component in the mixture
2. Apply EM until the log-likelihood  $\mathcal{L}$  or the parameters of the GMM converge (cf. Section 4.1)

The stopping criterion can either be a maximum pre-selected number of components, or it can be any other model selection criterion. In our case the algorithm stops if the maximum number of components is reached, or if the new model's log-likelihood  $\mathcal{L} + 1$  is less or equal to the log-likelihood  $\mathcal{L}$  of the previous model, after introducing a new component.

For a more formal description let us consider a feature vector set  $\mathcal{X}$  under a  $M$ -component mixture  $p^M(\mathcal{X}|\lambda)$ . The greedy learning procedure can be summarized in Algorithm 2. For each spatial feature vector, we estimate the parameters and the log-likelihood of an one-component model (Steps 4-5). In sequence, we find a new component and add it to the previous mixture (Steps 7-8). Then, we re-estimate the model parameters and log-likelihood (Steps 9-10) until we reach the desiderata described above.

---

### Algorithm 2: Optimized GMM Training

---

**Input:** A set of spatial feature vectors  $\hat{\mathcal{X}}$ , a maximum number of components in  $\mathcal{M}_c$   
**Output:** A set of trained GMMs  $\hat{\mathcal{G}}$

```

1 begin
2    $M \leftarrow 1$ 
3   foreach  $\mathcal{X} \in \hat{\mathcal{X}}$  do
4      $p^M(\mathcal{X}|\lambda) \leftarrow$  Estimate 1-component model parameters using EM
5      $\mathcal{L}^M \leftarrow$  Calculate 1-component model log-likelihood
6     while  $M \leq \mathcal{M}_c$  do
7        $g(\mathcal{X}; \lambda^*) \leftarrow$  Optimal new component for  $(p^M(\mathcal{X}|\lambda))$ 
8        $p^{M+1}(\mathcal{X}|\lambda) \leftarrow$  Combine model  $p^M(\mathcal{X}|\lambda)$  and component  $g(\mathcal{X}; \lambda^*)$  in a new model
9        $p^{M+1}(\mathcal{X}|\lambda) \leftarrow$  Estimate new model parameters using EM
10       $\mathcal{L}^{M+1} \leftarrow$  Calculate new model log-likelihood
11      if  $\mathcal{L}^{M+1} \leq \mathcal{L}^M$  then
12         $\hat{\mathcal{G}}.\text{PUSHGMM}(p^M(\mathcal{X}|\lambda))$ 
13        TERMINATE()
14      else
15         $M \leftarrow M + 1$ 
16      end
17    end
18     $\hat{\mathcal{G}}.\text{PUSHGMM}(p^M(\mathcal{X}|\lambda))$ 
19  end
20  return  $\hat{\mathcal{G}}$ 
21 end
```

---

The crucial step of this algorithm is the search for an optimal new component (Step 7). Several approaches exist for this issue: One is to consider a number of candidates equal to the number of feature vectors but it is identified that such strategy would be rather expensive. The approach followed in this work is to pick an optimal number of candidate components as discussed in [24]. More specifically, for each insertion problem in a  $k$ -component mixture, the dataset  $\hat{\mathcal{X}}$  is partitioned in  $k$  disjoint subsets and a fixed number  $m$  of candidate components is generated per existing mixture component, e.g., for a  $k$ -component mixture  $k \times m$  candidate components are generated. In our experiments we used  $m = 10$ . Finally, with the use of EM algorithm we pick the candidate component that maximizes the log-likelihood  $\mathcal{L} + 1$  when mixed into the previous mixture  $p^M(\mathcal{X}|\lambda)$ .

## 5 Location Fusion

Based on the discussion above, we introduce a location prediction system that employs the proposed spatial relation modeling approach described in Section 4. All the text processing parts are implemented in Python while modeling and experimentation

parts where implemented in Matlab. Our tests were conducted on an Intel(R) Core(TM) i5-2400 CPU at 3.10GHz with 8GB of RAM, running Ubuntu Linux 12.10.

### 5.1 Quantitative and Qualitative Performance

Our goal with this experiment is to show that our approach performs well for various location estimation scenarios using crowdsourced data. In particular, we assume that unknown locations in relation to known POIs do exist in crowdsourced data. The first scenario, e.g. random point location prediction, uses randomly generated POIs and tries to estimate their locations by fusing spatial relationship observations to known POIs (landmarks).

The procedure shown in the pseudocode of Algorithm 3 is as follows. At first, we partition the space with respect to grid vertices (landmarks) (Step 2). For example, for a grid dimensionality  $\mathcal{G}_D = 15$ , we have  $15 \times 15 = 225$  grid vertices and  $14 \times 14 = 196$  grid cells (regions). We generate a random point  $\mathcal{R}_P$  (Step 5) and, for each vertex (landmark)  $gv$ , starting with the bottom-left vertex and proceeding then to the right and in a row-by-row fashion, we find the spatial relation model  $\hat{\mathcal{G}}'$  that maximizes the likelihood—in terms of probability—of  $\mathcal{R}_P$  (Step 8). Then, the selected spatial relation model  $\hat{\mathcal{G}}'$  is used to assign positional probabilities to each other vertex (landmark)  $gv'$  of the grid using the current vertex as a reference point (Steps 10-13). In this way, we update the likelihoods of all vertices in the grid. All these likelihoods are stored in matrix  $\mathcal{V}_L$  (Step 11). Finally, all the likelihoods per vertex are summed up and normalized in  $\mathcal{F}_{V_L}$  (Step 16) and a probability is assigned to each grid cell (region)  $\mathcal{R}_L$  (Step 17). The overall likelihood of each grid cell is calculated as the mean value of the likelihoods of its four vertices. In a final step, we keep track of how many times the region that contains the unknown point is ranked among the  $K$ -highest (Top-K) probable regions (Steps 18-20). This allows us to measure the prediction accuracy of the proposed approach (Step 22).

---

#### Algorithm 3: Random Point Location Prediction

---

**Input:** A set of trained GMMs  $\hat{\mathcal{G}}$ , a bounding box  $\mathcal{B}_B$ , grid dimensionality in  $\mathcal{G}_D$ , a number  $\mathcal{R}_N$  of random points to be generated  
**Output:** Prediction accuracy  $\mathcal{A}$

```

1 begin
2    $\mathcal{G}_V \leftarrow$  Calculate grid vertices for  $\mathcal{B}_B$  based on  $\mathcal{G}_D$ 
3    $InTop5 \leftarrow 0$ 
4   for  $i \leftarrow 1$  to  $\mathcal{R}_N$  do
5      $\mathcal{R}_P \leftarrow$  Generate a random point in  $\mathcal{B}_B$ 
6      $Indx1 \leftarrow 0$ 
7     foreach  $gv \in \mathcal{G}_V$  do
8        $\hat{\mathcal{G}}' \leftarrow \arg \max_{g \in \hat{\mathcal{G}}} P(\mathcal{R}_P | g, gv)$ 
9        $Indx2 \leftarrow 0$ 
10      foreach  $gv' \in \mathcal{G}_V$  do
11         $\mathcal{V}_L(Indx1, Indx2) \leftarrow$  Calculate each  $gv'$  vertex's likelihood with  $gv$  as reference given model  $\hat{\mathcal{G}}'$ 
12         $Indx2 \leftarrow Indx2 + 1$ 
13      end
14       $Indx1 \leftarrow Indx1 + 1$ 
15    end
16     $\mathcal{F}_{V_L} \leftarrow$  Sum and normalize each vertex's likelihoods in  $\mathcal{V}_L$ 
17     $\mathcal{R}_L \leftarrow$  Calculate each region's likelihood using  $\mathcal{F}_{V_L}$ 
18    if  $\mathcal{R}_P$  in  $K$  highest probability  $\mathcal{R}_L$  then
19       $InTopK \leftarrow InTopK + 1$ 
20    end
21  end
22   $\mathcal{A} \leftarrow \text{PERCENT}(InTopK)$ 
23  return  $\mathcal{A}$ 
24 end

```

---

Figure 3 provides a more elaborate description of a simulation for very challenging case of Beijing. The light gray colored grid cell in Figure 3(a) illustrates the region in which a random point was generated. Figure 3(b) shows the assigned probabilities to each region after a full run of Algorithm 3. We observe that our approach assigns the highest probability for the random point location by using only spatial relation information extracted with respect to the landmarks in the region.

Figure 3(c) illustrates the monitoring of the log-likelihood of the random point region as we sequentially visit each vertex (landmark) in the grid. Starting from the lower left grid vertex ( $\hat{\mathcal{G}}'$  model 1) and proceeding row-wise until the upper right vertex ( $\hat{\mathcal{G}}'$  model 225), the log-likelihood increases as we move closer to the desired region and decreases when moving away.

Figure 3(d) points out the five highest likelihoods of the random point region and Figure 3(e) the locations of these vertices (landmarks) in the grid along with the model’s textual description. This analysis should be considered a qualitative accuracy assessment. Most of the models selected are qualitatively correct and express a real spatial relation between each corresponding vertex and random point. The relations shown in Figure 3(e) are of the form  $(P_u, R_o, P_v)$  as presented in Section 3.2.

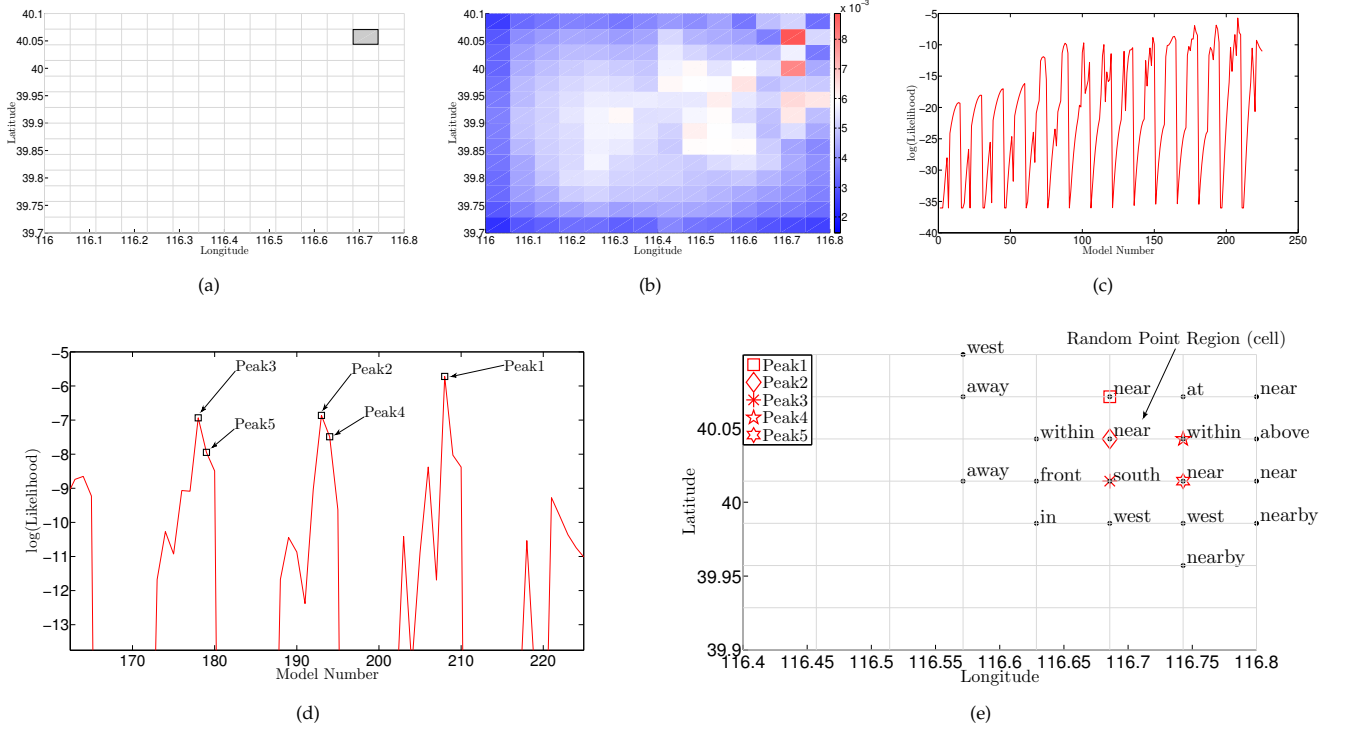


Figure 3: A location prediction scenario: (a) illustrates the region in which a random point has been generated, (b) visualizes using heatmap colors the probability of each region after a full run of Algorithm 3, (c) illustrates the log-Likelihood of the random point’s region as we traverse from one vertex (landmark) to the other, (d) shows an enlarged portion of (c) with the five highest likelihood peaks, and (e) illustrates the 20 best vertex-models and emphasizes the 5 best vertex-models (peaks in (d)).

To the best of our knowledge, this is the first work on location prediction based on textual descriptions. Consequently, in order to provide some comparison results, we define a 1 – component **baseline** (BSL) model (a GMM model  $p(x|\lambda) = \sum_{i=1}^M w_i g(x; \mu_i, \Sigma_i)$  with  $M = 1$ ), and an optimized model (OPT) trained as analyzed in Section 4.2 and Algorithm 2. We run Algorithm 3 for both BSL and OPT models, for all four datasets, with 1000 random points per dataset. Additionally, we consider the cases where the randomly generated point’s region is among the *TopK* predicted regions with  $K$  values 1, 5, 10, 20 respectively. The prediction accuracy results are shown in Figure 4. Figure 4(a) illustrates the prediction accuracy of the BSL model while Figure 4(b) illustrates the prediction accuracy of the OPT model. The results show the superiority of our model, as opposed to the 1-component model. Additionally, Table 2 shows the actual prediction accuracy improvement when we use the OPT model. In some cases (indicated in bold) the prediction accuracy improvement is equal to or greater than 30%. These results show that “colloquial” location estimation facilitated by crowdsourced geospatial narratives is a feasible approach.

Table 2: Prediction accuracy improvement when the optimized model (OPT) is used instead of the BSL model.

	Improvement per Top-k case			
Dataset	$k = 1$	$k = 5$	$k = 10$	$k = 20$
London	+ <b>34</b> %	+ <b>40</b> %	+16%	+15%
New York	+ <b>50</b> %	+ <b>53</b> %	+ <b>51</b> %	+ <b>50</b> %
Paris	+21%	+27%	+ <b>30</b> %	+29%
Beijing	+24%	+16%	+16%	+15%

Finally, we also want to measure the percentage of selected models  $\hat{\mathcal{G}}$  that are qualitatively correct, i.e., they reveal a true



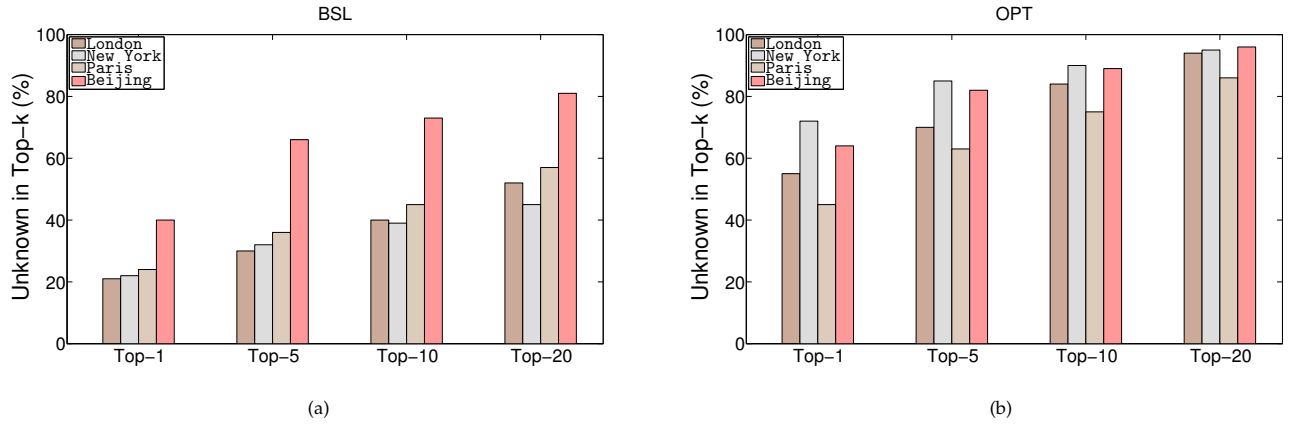


Figure 4: Location prediction accuracy. (a) Illustrates the prediction accuracy of the BSL model for  $K$  values 1, 5, 10, 20 respectively. (b) Illustrates the prediction accuracy of the OPT model for  $K$  values 1, 5, 10, 20 respectively.

spatial relation between a vertex and a random point. Figure 5 shows the percentage of the selected models  $\hat{\mathcal{G}}'$  that depict an accurate spatial relation between the vertices and random points for both BSL and OPT models. As in the prediction accuracy case, the qualitative accuracy of the OPT model is quite higher than that of the BSL model. Table 3 shows this improvement of OPT model over BSL model in relative terms. In some cases (indicated in bold) the qualitative accuracy improvement is more than 10%.

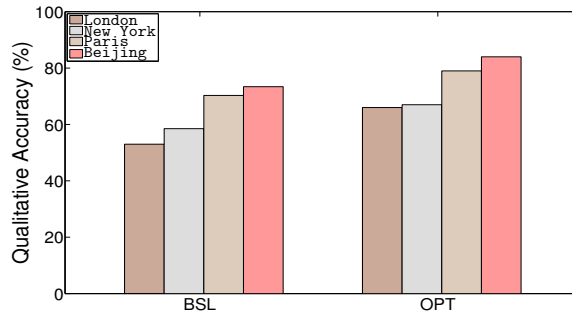


Figure 5: Illustrates the percentage of qualitatively correct spatial relations for the BSL and OPT models respectively.

Table 3: Qualitative accuracy improvement when the optimized model (OPT) is used instead of the BSL model.

Dataset	Improvement
London	+8%
New York	+ <b>11</b> %
Paris	+ <b>13</b> %
Beijing	+9%

To visualize the actual models and the respective probabilities they assign to partitioned space, Figure 6 depicts three instances of spatial relations, with the center of the grid denoting a reference (landmark) point. Figure 6 shows the spatial extend of relations (a) “Near”, (b) “At” and (c) “West” when searching for an unknown point that is spatially related to the center of the grid. The examples have been derived from the New York, London, and Beijing datasets. The concentration of measures around qualitatively correct regions is a further indication for the correctness of our models.

## 5.2 Real-world Location Estimation

The ultimate goal of this work is to train probabilistic models for spatial relationships so as to provide location estimation, e.g., finding the “best pita place in Greece”, based on hints in the form of qualitative spatial relationships discovered in textual

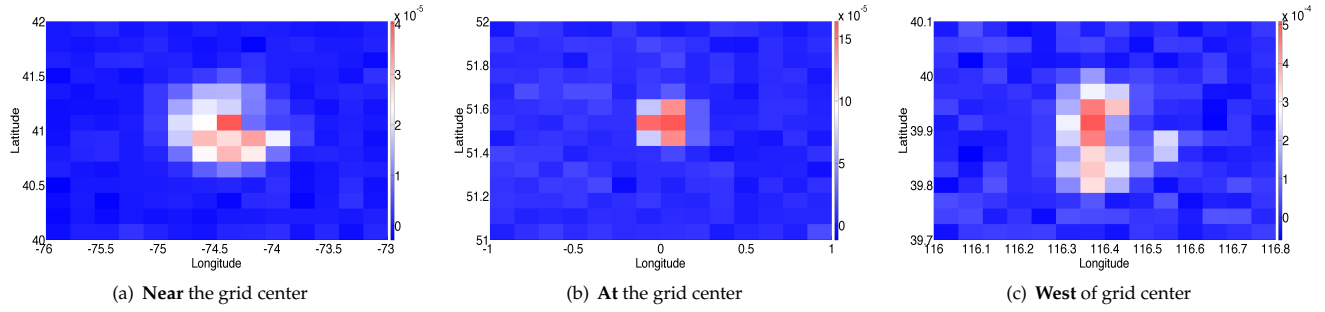


Figure 6: Spatial extension of spatial relationships - probabilities of specific spatial relationships (Near, At, West) relating vertices to the center grid cell.

narratives. This is a very important application as it provides a solution to the geocoding problem that exists on the *www*, i.e., there are millions of user referenced POIs whose coordinates do not exist in any coordinate database.

In addressing this challenge, we present four concrete location estimation scenarios. From our travel blog dataset, we extract four POIs (considered as unknown) whose locations are described in relation to other known POIs. Note that these cases have **not** been used in the training phase. When estimating the location of these POIs, we will also show the impact of an increased number of models on the quality of the location outcome. We start from a scenario where a POI is described by means of a few observations and subsequently increase this number.

Figure 7 illustrates the aforementioned scenarios. Figures 7(a), (b) and (c) illustrate an unknown POI (red star) in the greater area of London, whose position is described in relation to known POIs (black stars) using a total number of 15 spatial relations. Figure 7(a) shows the contours of the spatial probability distribution when only a randomly selected 50% of the spatial relations are taken into account, while Figure 7(b) shows the final distribution considering all spatial relations. Finally, Figure 7(c) is a closeup of Figure 7(b) with a GoogleMaps basemap overlay.

The same approach for assessing the impact of an increased number of models is used for the cases of New York, Beijing and Paris, with a total number of 20, 70 and 200 spatial relations being used in each case, respectively. Again, the unknown and known POIs are marked by a red and a black star, respectively.

What should be evident from these results is the considerable prediction accuracy. Especially the cases of Beijing (see Figures 7(g) 7(h) 7(i)) and Paris (see Figures 7(j) 7(k) 7(l)) *clearly pinpoint the unknown POI location*. What is further encouraging is that even for the cases of London (see Figures 7(a) 7(b) 7(c)) and New York (see Figures 7(d) 7(e) 7(f)) where the number of relations is small, the proposed approach works reasonably well.

As expected, the prediction accuracy increases with the number of observations (models) considered. This is confirmed by the mass of the probability moving closer to the unknown POI location when increasing the number of observations from a randomly selected 50% (Figure 7 1<sup>st</sup> column) to 100% (Figure 7 2<sup>nd</sup> column). This effect is observed for all four cases. Additionally, Table 4 shows the distances between the centers of the spatial probability distributions and the unknown POI locations as we increase the percentage of spatial relations considered for the prediction procedure. Here, we investigate the cases of 10%, 50% and 100% randomly selected relations.

Table 4: Distance between the center of the spatial probability distribution and the unknown POI.

Dataset	Percentage of relations considered		
	10%	50%	100%
London	15.3Km	7.9Km	7.7Km
New York	16.2Km	11.9Km	11.1Km
Beijing	14.4Km	8.6Km	<b>1.2Km</b>
Paris	8.7Km	<b>1.6Km</b>	<b>0.8Km</b>

The results show that as we increase the number of relations considered, we achieve more accurate estimates of the unknown POI, i.e., smaller distances between the estimated and the ground truth location. The improvement is considerable for all cases, with Beijing and Paris benefitting most and achieving distances of less than 2Km (indicated in bold in Table 4). Moreover, although the estimation quality (accuracy as well as precision) increases with the number of observations, nevertheless, even in the case of a small number of observations, we can rely on the crowd as a geospatial data source for location estimation.

We can conclude that the proposed modeling using GMMs optimized by the greedy EM algorithm presented in Section 4.2 can efficiently handle the uncertainty introduced by user-contributed qualitative geospatial data. In combination with information extraction techniques, it provides us with the non-trivial means of textual narrative-based location estimation.

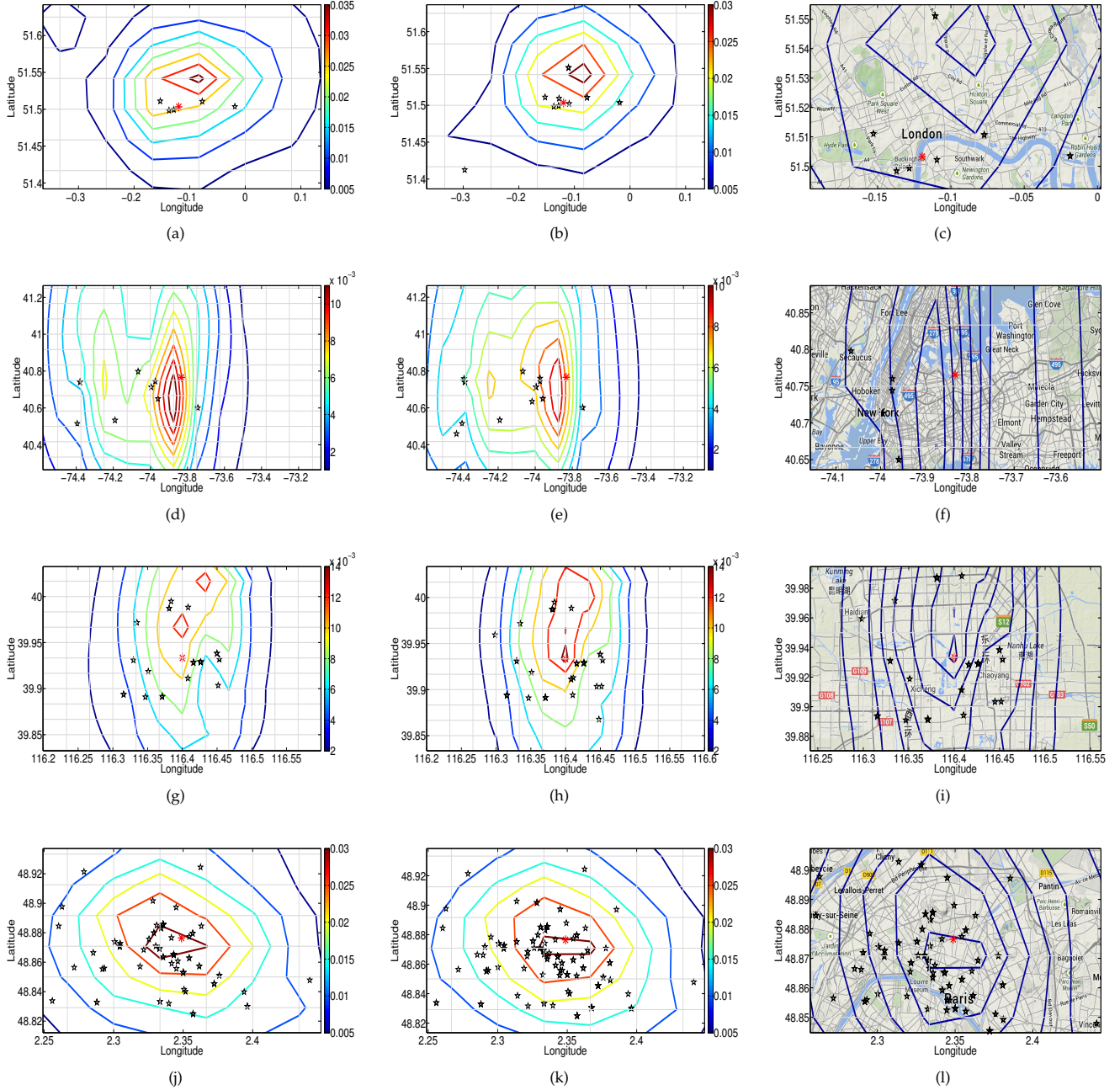


Figure 7: Real world location prediction scenarios - Rows 1 to 4 are scenarios for London, New York, Beijing, and Paris, respectively - Columns 1-3 are a randomly selected 50% , 100% and 100% (on GoogleMaps) of the discovered relations considered for prediction, respectively.

## 6 Conclusions

The increase in available user-generated data provides a unique opportunity for the generation of rich datasets in geographical information science. With textual narrative being the most popular form of human expression on the internet, this work provides a method that effectively translates this text into geospatial datasets. Our specific contribution is detecting spatial relationships in textual narratives and using them to “triangulate” the position of unknown objects. This is a first step for solving the emerging geocoding problem on the internet. We introduce specific techniques for extracting spatial relations from textual narratives and use a novel quantitative approach based on training probabilistic models for the representation of spatial relations. Combining these models and interpreting them as observations allows us to reason about unknown object locations. The proposed approach provides an optimized spatial relation modeling technique that achieves high-quality location estimation

results as evidenced by a range of real-world datasets. Here, our probabilistic approach is robust with respect to handling any uncertainties that characterize geospatial observations derived from crowd-sourced textual data.

Directions for future work include the optimization of the NLP techniques used for the automatic extraction of POIs and spatial relationship information from texts. Furthermore we will investigate the implementation of global prediction models, which could complement geocoding methods in our increasingly non-cartesian world. Also, this will enable us to evaluate additional probabilistic and deterministic modeling techniques and to develop more efficient text-to-map applications.

## References

- [1] S. Bird. Nltk: the natural language toolkit. In *Proc. of the COLING/ACL on Interactive presentation sessions*, pages 69–72.
- [2] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [3] R. Bunescu and R. Mooney. Subsequence Kernels for Relation Extraction. In *Advances in Neural Information Processing Systems 18*, pages 171–178.
- [4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.
- [5] E. Drymonas and D. Pfoser. Geospatial route extraction from texts. In *Proc. of the 1st ACM SIGSPATIAL Int’l Workshop on Data Mining for Geoinformatics*, pages 29–37, 2010.
- [6] M. Egenhofer. A formal definition of binary topological relationships. pages 457–472. 1989.
- [7] M. J. Egenhofer and J. Herring. A mathematical framework for the definitions of topological relationships. In *Int’l Symp. on Spatial Data Handling*.
- [8] M. J. Egenhofer and J. Sharma. Topological relations between regions in  $r^2$  and  $z^2$ . pages 316–336, 1993.
- [9] A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. In *Proc. of the Conf. of Empirical Methods in Natural Language Processing (EMNLP ’11)*.
- [10] R. H. Güting. An introduction to spatial database systems. *The VLDB Journal*, 3(4):357–399, 1994.
- [11] J. Hays and A. A. Efros. im2gps: estimating geographic information from a single image. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.
- [12] D. Hirschberg. Serial computations of levenshtein distances, 1997.
- [13] W. Kainz, M. J. Egenhofer, and I. Greasley. Modeling spatial relations and operations with partially ordered sets. *Int’l Journal of Geographical Information Systems*, 7:215–229, 1993.
- [14] D. V. Kalashnikov, Y. Ma, S. Mehrotra, R. Hariharan, and C. Butts. Modeling and querying uncertain spatial information for situational awareness applications. In *Proc. of the 14th annual ACM Int’l Symposium on Advances in Geographic Information Systems*, pages 131–138, 2006.
- [15] P. Kordjamshidi, M. van Otterlo, and M.-F. Moens. Spatial role labeling: Towards extraction of spatial relations from natural language. *ACM Transactions on Speech and Language Processing*, pages article 4, 36 p.
- [16] M. Koubarakis, T. K. Sellis, A. U. Frank, S. Grumbach, R. H. Güting, C. S. Jensen, N. A. Lorentzos, Y. Manolopoulos, E. Nardelli, B. Pernici, H.-J. Schek, M. Scholl, B. Theodoulidis, and N. Tryfona, editors. *Spatio-Temporal Databases: The Chorochronos Approach*, Lecture Notes in Computer Science, 2003.
- [17] J. Q. Li and A. R. Barron. Mixture density estimation. In *Advances in Neural Information Processing Systems 12*, pages 279–285. MIT Press.
- [18] Y. Ma, D. V. Kalashnikov, and S. Mehrotra. Toward managing uncertain spatial information for situational awareness applications. *IEEE Trans. on Knowl. and Data Eng.*, 20(10):1408–1423, 2008.
- [19] F. Mesquita, J. Schmidek, and D. Barbosa. Effectiveness and efficiency of open relation extraction. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing*, pages 447–457.
- [20] D. Papadias and T. Sellis. Qualitative representation of spatial knowledge in two-dimensional space. *The VLDB Journal*, 3(4):479–516, 1994.
- [21] D. Papadias, Y. Theodoridis, and T. Sellis. The retrieval of direction relations using r-trees, 1994.

- [22] G. Skoumas, D. Pfoser, and A. Kyrillidis. On quantifying qualitative geospatial data: A probabilistic approach. In *Proc. of the 2nd ACM SIGSPATIAL Int'l Workshop on Crowdsourced and Volunteered Geographic Information*, pages 71–78.
- [23] R. Smith, M. Self, and P. Cheeseman. Autonomous robot vehicles. pages 167–193. 1990.
- [24] J. J. Verbeek, N. Vlassis, and B. Kröse. Efficient greedy learning of gaussian mixture models. *Neural Computation*, 15:469–485, 2003.
- [25] Wanderlust. Extracting Semantic Relations from NaturalLanguage Text Using Dependency Grammar Patterns. *Proc. of the Workshop on Semantic Search (SemSearch 2009) at the 18th Int. World Wide Web Conf. (WWW 2009)*.
- [26] Y. Wang and F. Makedon. R-histogram: quantitative representation of spatial relations for similarity-based image retrieval. In *Procs. of the 11th ACM Int'l Conf. on Multimedia*, pages 323–326, 2003.
- [27] J. Xu and C. Yao. Formalizing natural-language spatial relations descriptions with fuzzy decision tree algorithm. *Proc. of Spie the Int'l Society for Optical Engineering*, 6420.
- [28] Y. Yuan. Extracting spatial relations from document for geographic information retrieval. In *2011 19th Int'l Conf. on Geoinformatics*, pages 1 –5.
- [29] D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relation extraction. *J. Mach. Learn. Res.*, pages 1083–1106.
- [30] X. Zhang, C. Zhang, C. Du, and S. Zhu. Svm based extraction of spatial relations in text. In *2011 IEEE Int'l Conf. on Spatial Data Mining and Geographical Knowledge Services (ICSDM)*, pages 529 –533.